MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

SUPPLEMENTARY EXAMINATION

TRIMESTER 1, 2015/2016

TCS 3311 – COMPILER DESIGN/ TCP2451 – PROGRAMMING LANGUAGE TRANSLATION

(All Sections / Groups)

18 NOV 2015
2.30 PM – 4.30 PM
(2 HOURS)

INSTRUCTION TO STUDENT

1. Answer **ALL** questions.
2. This question paper has 4 printed pages excluding the front cover.
3. Please print all your answers in the answer booklet provided.

QUESTION 1

Given the following context free grammar:

```
<vardecl>     ::= var <vardecllist> ;
<vardecllist> ::= <varandtype> { ; <varandtype> }
<varandtype>  ::= <ident> { , <ident> } : <typespec>
<ident>       ::= <letter> { <idchar> }
<idchar>      ::= <letter>  |  <digit>  |  _
<letter>    ::= a ... z | A ... Z
<digit>       ::= 0 | 1 | ... 9
<typespec>    ::= integer | boolean | real | char
```

(a) Create a token table for this grammar, giving all the token types and regular expressions to recognize each token.

[5 marks]

(b) Draw an NFA to recognize an identifier token based on the regular expression you derived in Question 1(a). Use Thompson's construction approach.

[4 marks]

(c) Convert the NFA in Question 1(b) to a DFA using subset construction method.

[6 marks]

Continued ...

QUESTION 2

Consider the following context free grammar:

$A \rightarrow B\ A'$
$A' \rightarrow +\ B\ A' \mid \varepsilon$

$B \rightarrow C\ B'$
$B' \rightarrow *\ D \mid /\ D$

$C \rightarrow r$
$D \rightarrow p \mid q$

(a) Write the FIRST set for all the non-terminals in the above grammar.

[5 marks]

(b) Write the FOLLOW set for all the non-terminals in the above grammar.

[5 marks]

(c) Draw a top-down parsing table for this grammar.

[5 marks]

Continued ...

QUESTION 3

(a) Consider the following context free grammar:

$S \rightarrow S + T \mid T$
$T \rightarrow (S) \mid id$

Draw the DFA to be used to build the LR(0) parser for this grammar.

[5 marks]

(b) Type checks, flow-of-control checks, uniqueness checks, and name-related checks are example of static checking done by a compiler. Elaborate each of the checks.

[5 marks]

(c) The Java Cup code below is incomplete. Complete the code so that an arithmetic expression can be parsed. The arithmetic expression works as one would expect – i.e. the arithmetic operators are evaluated as per mathematical evaluation order.

```
// Removed code goes here

expr ::= expr:e1 PLUS expr:e2
     {: RESULT = new Integer(e1.intValue() + e2.intValue()); :}
     | expr:e1 MINUS expr:e2
     {: RESULT = new Integer(e1.intValue() - e2.intValue()); :}
     | expr:e1 MULT expr:e2
     {: RESULT = new Integer(e1.intValue() * e2.intValue()); :}
     | expr:e1 DIV expr:e2
     {: RESULT = new Integer(e1.intValue() / e2.intValue()); :}
     | MINUS expr:e1 %prec UMINUS
     {: RESULT = new Integer(0 - e1.intValue(); :}
     | LPAREN expr:e1 RPAREN
     {: RESULT = e1; :}
     | NUMBER:n
     {: RESULT = n; :}
     ;
```

[5 marks]

Continued ...

QUESTION 4

(a) Global common subexpressions, copy propagation, dead-code elimination, and code motion are examples of global optimization techniques. Elaborate on them by giving examples.

[5 marks]

(b) What are the design issues in a code generator? Give 5 of them.

[5 marks]

(c) Discuss the scoping rules in a flat block structure.

[5 marks]

End of Page